



Second Generation Electronic Filing Specifications



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION

Court Policy 2.0 Version Test02

Staff and Consultants

Winchel "Todd" Vincent, III, Author
WTVIII, Inc. and <xmlLegal>
winchel@mindspring.com

Kevin Stannard, Editor
Swiftnet Solutions, Ltd.
kevin.stannard@swiftnet-solutions.com

Charlene Hammitt, Project Director
California Administrative Office of the Courts
charlene.hammitt@jud.ca.gov

Christopher Smith, Staff Analyst
California Administrative Office of the Courts
christopher.smith@jud.ca.gov

Tom Smith, Lead Consultant
AVI/ITDecisions
tjsmith@itdecision.com

Contributors

AVI/ITDecision

Tom Smith, tjsmith@itdecision.com

Counterclaim.com (<http://www.counterclaim.com>)

Shogan Naidoo shogan@counterclaim.com

Jim Beard beard@counterclaim.com

E-Filing.com (<http://www.e-filing.com>)

Mohammed Shaikh, mohammed@e-filing.com

Amrit Singh Nandrey, amrit@imagexx.com

Prabhath Pallati, prabhath@imagexx.com

Essential Publishers (<http://www.epubs.org>)

Martin Dean, mrsmary@epubs.org

ISD Corporation

Rob Beach, Ron.Beach@isd-corp.com

One Legal, Inc. (<http://www.onelegal.com>)

Robert DeFilippes, rtd@onelegal.com

Matt Marshall, matt@mattmarshall.com

Lexis-Nexis CourtLink (<http://www.lexisnexis.com>)

Shane Durham, shane.durham@lexisnexis.com

Sustain Technologies

Jon D. Salzman, jd.salzman@dailyjournal.com

Main Table of Contents

1.1.	Policy Schema	-3-
1.2.	CodeTable Schema	-11-
1.3.	Code Schema	-17-
1.4.	CourtKey Schema	-21-
1.5.	Exchange Schema	-24-
1.6.	HoursOfOperation Schema	-28-
1.7.	Payment Schema	-32-
2.0.	Legal Notices	-36-



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION



1.1. Policy Schema

Last Updated: 2003-07-27

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/>

Schema Prefix

Policy

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/Policy.xsd>

Table of Contents

1. Elements

Internal

1.1. Policy	-4-
1.2. Name	-5-
1.3. Version	-6-
1.4. PublicationDate	-6-
1.5. ExpirationDate	-6-
1.6. CourtKeys	-6-
1.7. CodeTables	-6-
1.8. Fees	-7-
1.9. Exchanges	-7-
1.10. Extensions	-7-

External

1.11. Key	-7-
1.12. CourtDetails	-7-
1.13. ClerkOfCourt	-7-
1.14. Fee	-7-
1.15. Payment	-7-
1.16. HoursOfOperation	-8-
1.17. Exchange	-8-
1.18. Extension	-8-

2. Simple Types

3. Imported Schemas

3.1. Attributes	-8-
3.2. Address	-8-
3.3. CourtDetails	-8-
3.4. Extension	-8-
3.5. Fee	-8-
3.6. Key	-8-
3.7. Organization	-8-
3.8. Person	-8-

3.9. CodeTable	-8-
3.10. CourtKey	-8-
3.11. Exchange	-8-
3.12. HoursOfOperation	-8-
3.13. Payment	-9-

4. Change History

4.1. 2003-02-18	-9-
4.2. 2003-02-23	-9-
4.3. 2003-04-29	-9-
4.4. 2003-06-20	-9-
4.5. 2003-07-01	-9-
4.6. 2003-07-02	-9-
4.7. 2003-07-23	-9-
4.8. 2003-07-26	-9-
4.9. 2003-07-27	-9-

1. Elements

1.1. Policy: Policy

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Key	1	1
Name	1	1
Version	1	1
PublicationDate	1	1
ExpirationDate	1	1
CourtDetails	0	1
CourtKeys	1	1
ClerkOfCourt	0	1
Fees	0	1
Payment	0	1
CodeTables	0	1
HoursOfOperation	0	1
Exchanges	0	1
Extensions	0	1

[1] This document is the [Court Policy 2.0 Specification](#). The web-based version is broken into several files, one for each schema. The Adobe PDF version is one PDF document for all schema unique to Policy and an additional PDF document that has documentation for "Building Block" schema. "Building Block" schema are common to both [Court Filing 2.0](#) and [Court Policy 2.0](#). Both the web-based format and the PDF format contain links to all schema documentation and the schema themselves. An accompanying document, [2GEFS Concepts](#), augments this specification with a high-level overview of terminology, models, and use cases.

[2] [Court Policy 2.0 \("CP 2.0"\)](#) includes information specific to a court and divisions or groups of a court, if such subdivisions exist. The purpose of [CP 2.0](#) is to encode, in a standard, machine-readable way, basic information about a court so that court filing applications can expect and use such information from many courts in the same way without the need to re-code. [CP 2.0](#) is, in other words, a court configuration file.

[3]Theoretically, CP 2.0 could contain a wide variety of information. Attempting to specify too much information in CP 2.0 is problematic because of the complexity and difficulty of writing code for all logical uses. As a result, CP 2.0 only includes the bare minimum of what could theoretically be in CP 2.0. The goal is to keep the XML as simple as possible to facilitate its use and adoption. Future versions may add additional information.

[4] CP 2.0 can include the following information about a Court: (a) unique names and identifiers for the court, its divisions, and its subdivisions, (b) court name and contact details, (c) clerk of court name and contact details, (d) fee schedules, (e) court payment details (enough information to pay a court), (f) code tables (frequently used code tables from the courts case management system(s), such as case categories), (g) electronic exchange points (names and unique identifiers for specific electronic information exchange points), (h) hours of operation for electronic filing, and (i) a means of specifying court policy extensions.

[5] CP 2.0 should be publicly available to authorized organizations, preferably over the Internet, for local or remote use. A standard means of publication is specified in this document, although implementing a standard means of publication is not necessary for the use of CP 2.0. Further, CP 2.0 may be downloaded and used locally, provided the Court Policy has not expired.

[6]The suggested method of publishing Court Policies is to publish policies on several mirrored web servers using the following directory structure. There is no requirement (or indeed a need) to use web services, as a simple filename at the end of a URI will suffice.

[7] Domain means any valid domain name. For example: <http://www.courtinfo.ca.gov/>.

[8] The directory CourtPolicies.

[9] : A directory using a two-letter ISO 3166 Country Code.

[10] : The fully spelled U.S. Postal Service State Name with no spaces and in UpperCamelCase.

[11] : The filename for the Court Policy, in the following format: OrganizationKey_YYYY_MM_DD_HH_MM_SS.xml, where OrganizationKey is a unique identifier issued to the court by a state Policy Authority; YYYY_MM_DD is the year, month, and day of publication; HH_MM_SS is the hour, minute, and second of publication; and .xml is the file extension.

[12]The OrganizationKey in the filename must match the OrganizationKey in the Court Policy itself. Further, the date in the filename must match the PublicationDate in the Court Policy itself. For example:

[13]The directory in which the Court Policies reside should have an xml file within it named index.xml. The index.xml should contain metadata about the files in the directory as defined by an XML Schema to be published during 2GEFS interoperability testing.

[14]Past Court Policies should be archived and indexed in the directory for a reasonable time.

[15] Policy is the intended root element of the schema.

1.2. Policy: Name

Data Type: xsd:string

[16] **Policy:Name** is a human readable name for the Court Policy. For example, **Court Policy for Douglas County Georgia, USA**. There is no special format for **Policy:Name**.

1.3. Policy: Version

Data Type: **xsd:string**

[17] **Policy:Version** is the version number for the policy. The suggested format for version numbers is **X.Y.Y** where **X** is a major version number and **Y.Y** is a minor version number. For example, **1.0.0**, **1.0.1**, **1.0.2** would be appropriate version numbers. Version numbers should be incremented by one minor version number for minor revisions and one major version number for major version changes. In case of a major revision, the minor version numbers should both be set to **0**. For example, **1.0.0** to **1.0.1** is a minor version change. **1.4.3** to **2.0.0** is a major version change.

[18] This version number should not be confused with the version number of the **CP 2.0** format. The **CP 2.0** version is determined by its namespace.

1.4. Policy: PublicationDate

Data Type: **xsd:date**

[19] **Policy:PublicationDate** is the date on which the policy was published or last updated. It is recommended that applications log this date when relying on the policy and that a copy of each version of each policy be kept by vendors or applications relying on the policy. Courts should archive old policies.

1.5. Policy: ExpirationDate

Data Type: **xsd:date**

[20] **Policy:ExpirationDate** is a future date on which the court determines the policy should expire. Applications should not rely on a policy after its expiration date. The **Policy:ExpirationDate** is useful so that applications can download and rely on local copies of a court policy.

[21] For example, if a court publishes its policy publicly on the Internet, an application can download it and use it locally until its expiration date. This is more efficient than checking the Internet location and downloading the policy for each use (which may be quite often). To reliably realize this efficiency, a court must not publish a new policy before the expiration date of its current policy. If it does so, then applications relying on local copies may be misinformed, since the application may reasonably rely on the local policy with a future **Policy:ExpirationDate** even though the court has (prematurely) published a new policy. It is suggested, therefore, that courts use short intervals between expiration dates, unless a court is reasonably certain that a policy is stable. A Court Policy should be republished with a new expiration date each time it expires.

1.6. Policy: CourtKeys

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
CourtKey:CourtKey	1	unbounded

[22] **Policy:CourtKeys** is a container element for multiple **CourtKey** values. A single court may have more than one **CourtKey** value for different court divisions, such as criminal, civil, or traffic. See **CourtKey** Schema.

1.7. Policy: CodeTables

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
CodeTable:CodeTable	1	unbounded

[23] **Policy:CodeTables** is a container element for multiple **CodeTable** elements. A single court is likely to have more than one **CodeTable** for each court and for different court divisions, such as criminal, civil, or traffic. See **CodeTable** Schema.

1.8. Policy: Fees

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Fee	1	unbounded

[24] **Policy:Fees** is a container element for multiple **Fee** elements. See **Fee** Schema.

1.9. Policy: Exchanges

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Exchange	1	unbounded

[25] **Policy:Exchanges** is a container element for multiple **Exchange** elements. An exchange is an Internet address where types of information exchanges occur to and from a court or justice agency. See **Exchange** Schema.

1.10. Policy: Extensions

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Extension	1	1

[26] **Policy:Extensions** is a container element for **Policy:Extension**. A **Policy:Extension** is a generic name/value pair that allows applications to provide information not included in the policy specification.

1.11. Policy: Key

Data Type: **Key:Key**

[27] **Policy:Key** is a unique identifier for a court policy. This specification provides a standard format for **Policy:Key** values. The **Policy:Key** format is the same as the **Filing:Key** format. See **Key** Schema.

1.12. Policy: CourtDetails

Data Type: **CourtDetails:CourtDetails**

[28] **Policy:Court** contains information about the court's name and contact details. See **Organization** Schema.

1.13. Policy: ClerkOfCourt

Data Type: **Person:Person**

[29] **Policy:ClerkOfCourt** includes information about the person who is the head clerk of the court. See **Person** Schema.

1.14. Policy: Fee

Data Type: **Fee:Fee**

[30] **Policy:Fee** contains information about court fees. See **Fee** Schema

1.15. Policy: Payment

Data Type: **Payment:Payment**

[31] **Policy:Payment** contains information about how to pay a court. See **Payment** Schema.

1.16. Policy: HoursOfOperation

Data Type: **HoursOfOperation:HoursOfOperation**

[32] **Policy:HoursOfOperation** contains information about the regular business days and hours on which the court is open, as well as holiday hours. See **HoursOfOperation** Schema.

1.17. Policy: Exchange

Data Type: **Exchange:Exchange**

[33] **Policy:Exchange** contains information about a court exchange. See **Exchange** Schema.

1.18. Policy: Extension

Data Type: **Extension:Extension**

[34]A **Policy:Extension** is a generic name/value pair that allows applications to send information not included in the policy specification. See **Policy:Extensions**.

2. Simple Types

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

3.2. Address

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Address/01/>

3.3. CourtDetails

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/CourtDetails/01/>

3.4. Extension

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Extension/01/>

3.5. Fee

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Fee/01/>

3.6. Key

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Key/01/>

3.7. Organization

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Organization/01/>

3.8. Person

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Person/01/>

3.9. CodeTable

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/CodeTable/01/>

3.10. CourtKey

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/CourtKey/01/>

3.11. Exchange

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/Exchange/01/>

3.12. HoursOfOperation

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/HoursOfOperation/01/>

3.13. Payment

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/Payment/01/>

4. Change History

4.1. 2003-02-18

Editor: Winchel Vincent

Added Structured Copyright and Update History

4.2. 2003-02-23

Editor: Winchel Vincent

Changed documentation annotations from complexTypes to elements so they would show up in XML Spy

4.3. 2003-04-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Policy/01/>

Changed Court element to CourtDetails. Add CourtDetails module. Added Extensions element and Extension module.

4.4. 2003-06-20

Editor: Winchel Vincent

Fixed relative path in Attribute import statement.

4.5. 2003-07-01

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Policy/Test01/>

Copied. Added HoursOfOperation. Added Key schema.

4.6. 2003-07-02

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.7. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test01/>

Copied

4.8. 2003-07-26

Editor: Winchel Vincent

Changed Key, CourtDetails, and Extension namespaces and import locations to reflect move of schema to Building Block folder.

4.9. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION



1.2.CodeTable Schema

Last Updated: 2003-07-27

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/CodeTable/01/>

Schema Prefix

CodeTable

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/CodeTable/01/CodeTable.xsd>

Table of Contents

1. Elements

Internal

1.1. CodeTable	-12-
1.2. Codes	-12-
1.3. Name	-12-
1.4. Alias	-12-
1.5. CourtKey	-13-
1.6. Mappings	-13-
1.7. Mapping	-13-
1.8. Filters	-13-
1.9. Filter	-13-
1.10. FilterName	-14-
1.11. FilterValue	-14-
1.12. Specification	-14-
1.13. ElementName	-14-
1.14. XPathLocation	-14-

External

1.15. Code	-14-
------------	------

2. Simple Types

2.1. YesNo	-14-
2.2. Uses	-14-

3. Imported Schemas

3.1. Attributes	-15-
3.2. Code	-15-

4. Change History

4.1. 2003-02-18	-15-
4.2. 2003-02-23	-15-
4.3. 2003-04-29	-15-
4.4. 2003-07-01	-15-
4.5. 2003-07-02	-15-
4.6. 2003-07-16	-15-

4.7. 2003-07-23

-15-

4.8. 2003-07-27

-15-

1. Elements

1.1. CodeTable: CodeTable

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Name	1	1
Alias	0	1
CourtKey	1	1
Mappings	0	1
Filters	0	1
Codes	1	1

Attribute(s)	type	use	fixed/default
Alphabetize	YesNo	optional	None
Use	Uses	optional	None

[35] **CodeTable** is the intended root element of the schema. **CodeTable** contains values for court case management system code tables. Tables might include case categories or document types.

[36]The attribute is used to suggest to an application whether to alphabetize the list when presenting it to users or whether to keep the values in document order. The attribute is only a suggestion. An application may ignore the attribute.

[37]The attribute tells an application whether the code table is required, depreciated, optional, or prohibited.

1.2. CodeTable: Codes

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Code	1	unbounded

[38] **CodeTable:Codes** is a container element for one or more codes in a code table.

1.3. CodeTable: Name

Data Type: xsd:string

[39] **CodeTable:Name** is the exact string value of the code table name in the court case management system. This string is used to match the Court Policy **CodeTable:Name** with the case management system table name.

1.4. CodeTable: Alias

Data Type: xsd:string

[40] **CodeTable:Alias** is an optional element that can be used to map the court case management system table name (e.g., **CodeTable:Name**) to either (a) a human readable name or (b) to a standard table name adopted for multiple case management systems in the jurisdiction. Use of **CodeTable:Alias** in situation (b) is recommended for harmonizing code table names on a state or federal level.

1.5. CodeTable: CourtKey

Data Type: **xsd:string**

[41] **CodeTable:CourtKey** is a value for a **CourtKey** for the court or a court division listed in the **Court Policy**. For example, if Superior Court is divided into criminal and criminal divisions, then one court policy will have two **CourtKeys**, one for each division of the court. In this case, **CodeTable:CourtKey** must be used to distinguish code tables used for each of the courts. If there are two code tables that are exactly the same (such as a code table for **Salutations**), then the code table must be duplicated using each of the **CourtKeys** corresponding to the court in which the code table is used.

1.6. CodeTable: Mappings

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Mapping	1	unbounded

[42] **CodeTable:Mappings** is a container element for the **CodeTable:Mapping** element. See **CodeTable:Mapping** element for more information.

1.7. CodeTable: Mapping

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Specification	1	1
ElementName	1	1
XPathLocation	1	unbounded

[43] **CodeTable:Mapping** is used to map code tables to other XML specifications. For example, if a Court Policy Code Table value is **Case Categories**, then information in the **CodeTable:Mapping** element can be used to map to the **CaseCategory** element in **Legal XML Court Filing 1.0**.

1.8. CodeTable: Filters

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Filter	1	unbounded

[44] **CodeTable:Filters** is a container element for the **CodeTable:Filter** element. See **CodeTable:Filter** element for more information.

1.9. CodeTable: Filter

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
FilterName	1	1
FilterValue	1	1

[45]The selection of values in one Code Table may create dependencies that limit the logical use of values in another table. Although representing such dependencies could be done with a longer **CourtKey**, it would make the **CourtKey** very long. Further, the **CourtKey** would no longer represent a physical/logical division of the court, but rather a case type (or something else) of a division within a court. This is undesirable. **CodeTable:Filter**, therefore, is used to filter tables based on values in other tables.

[46]Based on the assumptions that Code Table dependencies (a) are complex, varied, and specific to an

implementation and (b) do not always occur, the Court Policy filter mechanism uses simple, generic **CodeTable:Filter** elements. Each Code Table or Code may have one or more **CodeTable:Filter** elements. Each **CodeTable:Filter** has a **CodeTable:FilterName** and **CodeTable:FilterValue**. Either the Court or a Service Provider may create its own filters by manipulating a local Court Policy. The Name/Value pair may be any name and value.

1.10. CodeTable: FilterName

Data Type: **xsd:string**

[47] **CodeTable:FilterName** is the name of a code table filter. See **CodeTable:Filter**.

1.11. CodeTable: FilterValue

Data Type: **xsd:string**

[48] **CodeTable:FilterValue** is the value of a code table filter. See **CodeTable:Filter**.

1.12. CodeTable: Specification

Data Type: **xsd:string**

[49] **CodeTable:Specification** is a unique name of the specification to which a code table is intended to be mapped.

1.13. CodeTable: ElementName

Data Type: **xsd:string**

[50] **CodeTable:ElementName** is the name of the element in the specification being mapped.

1.14. CodeTable: XPathLocation

Data Type: **xsd:string**

[51] **CodeTable:XPathLocation** is a valid XPath string that points to the element in the specification being mapped.

1.15. CodeTable: Code

Data Type: **Code:Code**

[52] **CodeTable:Code** is an individual code in a code table. See **Code** Schema.

2. Simple Types

2.1. YesNo

Data Type: **xsd:string**

Enumeration(s)
Yes
No

2.2. Uses

Data Type: **xsd:string**

Enumeration(s)
Required
Optional
Deprecated

Prohibited

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

3.2. Code

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/CodeTable/Code/01/>

4. Change History

4.1. 2003-02-18

Editor: Winchel Vincent

Added Structured Copyright and Update History

4.2. 2003-02-23

Editor: Winchel Vincent

Changed documentation annotations from complexTypes to elements so they would show up in XML Spy

4.3. 2003-04-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Policy/01/CodeTable/01/>

4.4. 2003-07-01

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Policy/Test01/CodeTable/01/>

Copied. Added Filters and Filter elements.

4.5. 2003-07-02

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.6. 2003-07-16

Editor: Winchel Vincent

Corrected incorrect schemaLocation paths.

4.7. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test01/CodeTable/01/>

Copied

4.8. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION



1.3.Code Schema

Last Updated: 2003-07-27

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/CodeTable/Code/01/>

Schema Prefix

Code

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/CodeTable/Code/01/Code.xsd>

Table of Contents

1. Elements

Internal

1.1. Code	-17-
1.2. Name	-18-
1.3. Value	-18-
1.4. Alias	-18-
1.5. Filters	-18-
1.6. Filter	-18-
1.7. FilterName	-19-
1.8. FilterValue	-19-
1.9. Include	-19-

External

2. Simple Types

2.1. YesNo	-19-
------------	------

3. Imported Schemas

3.1. Attributes	-20-
3.2. Organization	-20-

4. Change History

4.1. 2003-02-18	-20-
4.2. 2003-02-23	-20-
4.3. 2003-04-29	-20-
4.4. 2003-07-01	-20-
4.5. 2003-07-02	-20-
4.6. 2003-07-16	-20-
4.7. 2003-07-23	-20-
4.8. 2003-07-27	-20-

1. Elements

1.1. Code: Code

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Name	1	1
Value	1	1
Alias	0	1
Filters	0	1
Include	1	1

[53] **Code** is the intended root element of the schema. The **Code** element contains information about a specific code in a code table. Most courts use a string value plus a numeric or alphanumeric code value, so **Code** has both **Code:Name** and **Code:Value** element. Code also has a **Code:Alias** element so other values can be mapped to **Code:Name** values.

1.2. Code: Name

Data Type: **xsd:string**

[54] **Code:Name** is a human-readable name associated with the code value taken from a case management system code table.

1.3. Code: Value

Data Type: **xsd:string**

[55] **Code:Value** is a numeric or alphanumeric code value taken from a case management system code table.

1.4. Code: Alias

Data Type: **xsd:string**

[56] **Code:Alias** is an alias for the **Code:Name**. The **Code:Alias** can be used to (a) provide a more user friendly code name or (b) to map a **Code:Name** or **Code:Value** to a standard set of codes harmonized and used in one jurisdiction for many courts. For example, Court A might use **Motion: Summary Judgment** as a **Code:Name**, while Court B might use **Motion for Summary Judgment** as a **Code:Name**. If the State standardizes on the string **Motion for Summary Judgment**, Court A can use **Code:Alias** to map its **Code:Name** to the State's standard code name. If values for both **Code:Name** and **Code:Alias** are present, then applications should present users with **Code:Alias** rather than **Code:Name**.

1.5. Code: Filters

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Filter	1	unbounded

[57] **Code:Filters** is a container element for the **Code:Filter** element. See **Code:Filter** element for more information.

1.6. Code: Filter

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
FilterName	1	1
FilterValue	1	1

[58] The selection of values in one Code may create dependencies that limit the logical use of values in another table. Although representing such dependencies could be done with a longer **CourtKey**, it would make the **CourtKey** very long. Further, the **CourtKey** would no longer represent a physical/logical division of the court, but

rather a case type (or something else) of a division within a court. This is undesirable. `CodeTable:Filter`, therefore, is used to filter tables based on values in other tables.

[59]Based on the assumptions that Code dependencies (a) are complex, varied, and specific to an implementation and (b) do not always occur, the Court Policy filter mechanism uses simple, generic `Code:Filter` elements. Each Code Table or Code may have one or more `Code:Filter` elements. Each `Code:Filter` has a `Code:FilterName` and `Code:FilterValue`. Either the Court or a Service Provider may create its own filters by manipulating a local Court Policy. The Name/Value pair may be any name and value.

1.7. Code: FilterName

Data Type: `xsd:string`

[60] `Code:FilterName` is the name of a code table filter. See `Code:Filter`.

1.8. Code: FilterValue

Data Type: `xsd:string`

[61] `Code:FilterValue` is the value of a code table filter. See `Code:Filter`.

1.9. Code: Include

Data Type: `YesNo`

[62] `Code:Include` provides a simple filter mechanism. The `Code:Include` element provides a `Yes/No` value for each `Code Value/Name/Alias`. This allows either a Court or a Service Provider to decide whether to include or not include a Code for use in an application.

[63]The `Code:Include` element functions based on the assumption that the following is allowed (a) a Court publishes basic policy with Code values in it, and (b) an application is allowed to download a policy locally, alter the policy, and create one or more derivative policies for its own internal uses. This means that while there is always one officially published Court Policy maintained by the Court or by a Policy Authority, local court policies can be manipulated for specific applications.

[64]For example, assume a Court Policy has 100 `Filer Types` in it. However, the Court Policy, in a particular installation, only requires one of the `Filer Types` (for example, the value `Sheriff/Law Enforcement`). In this situation, the Sheriff, a Filer, does not need to see all of the possible `Filer Types`, only the `Sheriff/Law Enforcement` Filer Type. In this case, in the local Court Policy used by the Sheriff, the Include value for `Sheriff/Law Enforcement` could be set to `Yes` and all other values could be set to `No`. The application would be required, in this case, to only show the Code values matching Include elements with `Yes` values. This provides a simple means of filtering values and also ensures that the Filer will not select the wrong value.

2. Simple Types

2.1. YesNo

Data Type: `xsd:string`

Enumeration(s)
[Blank Value]
Yes
No

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

3.2. Organization

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Organization/01/>

4. Change History

4.1. 2003-02-18

Editor: Winchel Vincent

Added Structured Copyright and Update History

4.2. 2003-02-23

Editor: Winchel Vincent

Changed documentation annotations from complexTypes to elements so they would show up in XML Spy

4.3. 2003-04-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Policy/01/CodeTable/Code/01/>

4.4. 2003-07-01

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Policy/Test01/CodeTable/Code/01/>

Copied. Added Filters and Filter elements.

4.5. 2003-07-02

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.6. 2003-07-16

Editor: Winchel Vincent

Corrected incorrect schemaLocation paths.

4.7. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test01/CodeTable/Code/01/>

Copied

4.8. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION



1.4.CourtKey Schema

Last Updated: 2003-07-27

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/CourtKey/01/>

Schema Prefix

CourtKey

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/CourtKey/01/CourtKey.xsd>

Table of Contents

1. Elements

Internal

1.1. CourtKey	-21-
1.2. Key	-22-
1.3. Division	-22-
1.4. Group	-22-
1.5. Include	-22-

External

2. Simple Types

2.1. YesNo	-22-
-------------------	------

3. Imported Schemas

3.1. Attributes	-22-
------------------------	------

4. Change History

4.1. 2003-02-18	-22-
4.2. 2003-02-23	-22-
4.3. 2003-04-29	-23-
4.4. 2003-06-20	-23-
4.5. 2003-07-01	-23-
4.6. 2003-07-02	-23-
4.7. 2003-07-16	-23-
4.8. 2003-07-23	-23-
4.9. 2003-07-27	-23-

1. Elements

1.1. CourtKey: CourtKey

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Key	1	1

Division	0	1
Group	0	1
Include	0	1

[65] **CourtKey** is the intended root element of the schema.

1.2. CourtKey: Key

Data Type: **xsd:string**

[66] **CourtKey:Key** is a unique identifier that identifies a court or a subdivision of the court.

1.3. CourtKey: Division

Data Type: **xsd:string**

[67] **CourtKey:Division** is the division of the court identified.

1.4. CourtKey: Group

Data Type: **xsd:string**

[68] **CourtKey:Group** is a group identified within a division of the court

1.5. CourtKey: Include

Data Type: **YesNo**

[69] **CourtKey:Include** is a flag that courts can use to tell applications whether the **CourtKey** is public or not.

2. Simple Types

2.1. YesNo

Data Type: **xsd:string**

Enumeration(s)
[Blank Value]
Yes
No

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

4. Change History

4.1. 2003-02-18

Editor: Winchel Vincent

Added Structured Copyright and Update History

4.2. 2003-02-23

Editor: Winchel Vincent

Changed documentation annotations from complexTypes to elements so they would show up in XML Spy

4.3. 2003-04-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Policy/01/CourtKey/01/>

4.4. 2003-06-20

Editor: Winchel Vincent

Fixed relative path in Attribute import statement.

4.5. 2003-07-01

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Policy/Test01/CourtKey/01/>

Copied

4.6. 2003-07-02

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.7. 2003-07-16

Editor: Winchel Vincent

Corrected incorrect schemaLocation paths.

4.8. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test01/CourtKey/01/>

Copied

4.9. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION



1.5.Exchange Schema

Last Updated: 2003-07-27

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/Exchange/01/>

Schema Prefix

Exchange

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/Exchange/01/Exchange.xsd>

Table of Contents

1. Elements

Internal

1.1. Exchange	-24-
1.2. Key	-25-
1.3. Name	-25-
1.4. CourtKey	-25-
1.5. Address	-25-
1.6. Include	-25-

External

2. Simple Types

2.1. YesNo	-25-
------------	------

3. Imported Schemas

3.1. Attributes	-26-
-----------------	------

4. Change History

4.1. 2003-02-18	-26-
4.2. 2003-02-23	-26-
4.3. 2003-04-29	-26-
4.4. 2003-06-20	-26-
4.5. 2003-07-01	-26-
4.6. 2003-07-02	-26-
4.7. 2003-07-16	-26-
4.8. 2003-07-23	-26-
4.9. 2003-07-27	-27-

1. Elements

1.1. Exchange: Exchange

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
-----------------	-----------	-----------

Key	0	1
Name	0	1
CourtKey	1	1
Address	1	1
Include	0	1

[70] **Exchange** contains information about Internet addresses where Exchange Policies are located. Exchange Policies can be used to configure exchange points for electronic information.

1.2. Exchange: Key

Data Type: **xsd:string**

[71] **Exchange:Key** is a unique identifier for the exchange.

1.3. Exchange: Name

Data Type: **xsd:string**

[72] **Exchange:Name** is a human readable name for the exchange. For example, **Douglas County Superior Court Electronic Court Filing Exchange Policy**.

1.4. Exchange: CourtKey

Data Type: **xsd:string**

[73] **Exchange:CourtKey** is a value for a **CourtKey** for the court or a court division listed in the Court Policy. For example, if a Superior Court is divided into criminal and criminal divisions, then one court policy will have two CourtKeys, one for each division of the court. In this case, **Exchange:CourtKey** must be used to distinguish exchanges used for each of the court divisions. If there are two exchanges that have the same **Exchange:Address**, then the exchange must be duplicated using each of the **CourtKeys** corresponding to the court in which the exchange is used.

1.5. Exchange: Address

Data Type: **xsd:string**

[74] **Exchange:Address** is the Internet address (URI) of the Exchange Policy. **Exchange:Address** is not the Internet address (URI) of the 'Exchange Point.' The Exchange Policy is a policy, encoded in XML as a Court Policy, that describes an Exchange Point. An Exchange Point is an Internet address (URI) where the actual electronic information exchange takes place.

1.6. Exchange: Include

Data Type: **YesNo**

[75] **Exchange:Include** is a flag that tells an application whether to include the Exchange in a public list or not.

2. Simple Types

2.1. YesNo

Data Type: **xsd:string**

Enumeration(s)
Yes
No

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

4. Change History

4.1. 2003-02-18

Editor: Winchel Vincent

Added Structured Copyright and Update History

4.2. 2003-02-23

Editor: Winchel Vincent

Changed documentation annotations from complexTypes to elements so they would show up in XML Spy

4.3. 2003-04-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Policy/01/Exchange/01/>

4.4. 2003-06-20

Editor: Winchel Vincent

Fixed relative path in Attribute import statement.

4.5. 2003-07-01

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Policy/Test01/Exchange/01/>

Copied

4.6. 2003-07-02

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9. Removed blank enumeration for YesNo.

4.7. 2003-07-16

Editor: Winchel Vincent

Corrected incorrect schemaLocation paths.

4.8. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test01/Exchange/01/>

Copied

4.9. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION



1.6.HoursOfOperation Schema

Last Updated: 2003-07-27

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/HoursOfOperation/01/>

Schema Prefix

HoursOfOperation

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/HoursOfOperation/01/HoursOfOperation.xsd>

Table of Contents

1. Elements

Internal

1.1. HoursOfOperation	-28-
1.2. Days	-29-
1.3. Holidays	-29-
1.4. DayName	-29-
1.5. Open	-29-
1.6. Close	-29-
1.7. HolidayName	-29-
1.8. Note	-30-
1.9. Date	-30-

External

2. Simple Types

2.1. DayNames	-30-
2.2. Memo	-30-

3. Imported Schemas

3.1. Attributes	-30-
-----------------	------

4. Change History

4.1. 2003-07-01	-30-
4.2. 2003-07-02	-30-
4.3. 2003-07-23	-30-
4.4. 2003-07-26	-31-
4.5. 2003-07-27	-31-

1. Elements

1.1. HoursOfOperation: HoursOfOperation

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
-----------------	-----------	-----------

Days	1	7
Holidays	1	unbounded

[76] **HoursOfOperation** is the intended root element of the schema. **HoursOfOperation** contains elements necessary to state the regular business hours of a court as well as holiday hours.

1.2. HoursOfOperation: Days

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
DayName	1	1
Open	1	1
Close	1	1

[77] **HoursOfOperation:Days** is container element for **HoursOfOperation:DayName** elements used to state the court's regular business hours for each day of the week. Information about specific days (e.g., **December 31st, 2003**) is stated in the **HoursOfOperation:Holidays** element.

1.3. HoursOfOperation: Holidays

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
HolidayName	1	1
Date	1	1
Open	1	1
Close	1	1
Note	1	1

[78] **HoursOfOperation:Holidays** is a container element for elements used to state a court's holiday hours of operation.

1.4. HoursOfOperation: DayName

Data Type: **DayNames**

[79] **HoursOfOperation:DayName** is an element that contains one of seven days of the week, e.g., **Monday**, **Tuesday**, etc.

1.5. HoursOfOperation: Open

Data Type: **xsd:time**

[80] **HoursOfOperation:Open** is the time at which the court opens on a particular day.

1.6. HoursOfOperation: Close

Data Type: **xsd:time**

[81] **HoursOfOperation:Close** is the time at which the court closes on a particular day.

1.7. HoursOfOperation: HolidayName

Data Type: **xsd:string**

[82] **HoursOfOperation:HolidayName** is a human-readable name or description of the name of the holiday, e.g., **Thanksgiving**.

1.8. HoursOfOperation: Note

Data Type: **Memo**

[83] **HoursOfOperation:Note** is a special note added to Holiday hours of operation.

1.9. HoursOfOperation: Date

Data Type: **xsd:date**

[84] **HoursOfOperation:Date** is the specific day and date on which a holiday falls.

2. Simple Types

2.1. DayNames

Data Type: **xsd:string**

Enumeration(s)
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

2.2. Memo

Data Type: **xsd:string**

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

4. Change History

4.1. 2003-07-01

Editor: Winchel Vincent

Created

4.2. 2003-07-02

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.3. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test01/HoursOfOperation/01/>

Copied

4.4. 2003-07-26

Editor: Winchel Vincent

Changed Date.xsd to xsd:date.

4.5. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION



1.7.Payment Schema

Last Updated: 2003-07-27

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/Payment/01/>

Schema Prefix

Payment

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test02/Payment/01/Payment.xsd>

Table of Contents

1. Elements

Internal

1.1. Payment	-32-
1.2. BankName	-33-
1.3. AccountName	-33-
1.4. AccountNumber	-33-
1.5. RoutingNumber	-33-
1.6. SwiftCode	-33-
1.7. BSBNumber	-33-
1.8. Instructions	-33-

External

1.9. BankAddress	-33-
-------------------------	------

2. Simple Types

2.1. Memo	-34-
------------------	------

3. Imported Schemas

3.1. Attributes	-34-
3.2. Address	-34-

4. Change History

4.1. 2003-02-18	-34-
4.2. 2003-02-23	-34-
4.3. 2003-04-29	-34-
4.4. 2003-06-20	-34-
4.5. 2003-07-01	-34-
4.6. 2003-07-02	-34-
4.7. 2003-07-23	-35-
4.8. 2003-07-27	-35-

1. Elements

1.1. Payment: Payment

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
BankName	0	1
BankAddress	0	1
AccountName	0	1
AccountNumber	0	1
RoutingNumber	0	1
SwiftCode	0	1
BSBNumber	0	1
Instructions	0	1

[85] **Payment** contains the court payment information, necessary for a vendor or other collection agency to pay the court for filing fees or other amounts collected on the court's behalf. Payment information is not intended to be directly provided to filers unless the court gives permission.

1.2. Payment: BankName

Data Type: **xsd:string**

[86] **Payment:BankName** is the name of the court's bank.

1.3. Payment: AccountName

Data Type: **xsd:string**

[87] **Payment:AccountName** is the name of the court's bank account.

1.4. Payment: AccountNumber

Data Type: **xsd:string**

[88] **Payment:AccountNumber** is the court's bank account number.

1.5. Payment: RoutingNumber

Data Type: **xsd:string**

[89] **Payment:RoutingNumber** is the routing number for the court's bank.

1.6. Payment: SwiftCode

Data Type: **xsd:string**

[90] **Payment:SwiftCode** is the Swift Code number, if any.

1.7. Payment: BSBNumber

Data Type: **xsd:string**

[91] **Payment:BSBNumber** is the BSBNumber, if any.

1.8. Payment: Instructions

Data Type: **Memo**

[92] **Payment:Instructions** are any human-readable instructions sent with the payment transaction.

1.9. Payment: BankAddress

Data Type: **Address:Address**

[93] **Payment:BankAddress** is the court's bank's address.

2. Simple Types

2.1. Memo

Data Type: **xsd:string**

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

3.2. Address

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Address/01/>

4. Change History

4.1. 2003-02-18

Editor: Winchel Vincent

Added Structured Copyright and Update History

4.2. 2003-02-23

Editor: Winchel Vincent

Changed documentation annotations from complexTypes to elements so they would show up in XML Spy

4.3. 2003-04-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Policy/01/Payment/01/>

4.4. 2003-06-20

Editor: Winchel Vincent

Fixed relative path in Attribute schema import statement.

4.5. 2003-07-01

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Policy/Test01/Payment/01/>

Copied

4.6. 2003-07-02

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9. Corrected Attributes and Address namespace declarations and import statements.

4.7. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Policy/Test01/Payment/01/>

Copied

4.8. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

2.0 Legal Notices

Unless otherwise agreed, All Rights Reserved except those granted by xmlLegal General Public License at:

<http://www.xmllegal.org/Legal/GeneralPublicLicense.htm>

LICENSED WORKS ARE PROVIDED "AS IS," AND HOLDERS OF INTELLECTUAL PROPERTY RIGHTS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE LICENSED WORKS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

HOLDERS OF INTELLECTUAL PROPERTY WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE LICENSED WORKS.

Contact Winchel "Todd" Vincent III (winchel@mindspring.com) or XMLLegal Help (help@xmllegal.org) for more information